

Amendments to the Claims

Kindly amend claims 1, 3 & 5 and cancel claims 7, 13 & 19 (without prejudice) as set forth below. All pending claims are reproduced below, with changes in the amended claims shown by underlining (for added matter) and strikethrough/double brackets (for deleted matter).

1. (Currently Amended) A method of managing locking of resources of a global data repository of a distributed computing environment, said method comprising:

issuing a request, via a thread of a multithreaded client application of said distributed computing environment, for a lock of a resource associated with a server data tree of said global data repository; and

obtaining said lock for said ~~thread~~ multithreaded client application, said lock being independent of a threading model of ~~an operating system~~ the requesting thread of the multithreaded client application of said distributed computing environment,

wherein said obtaining comprises employing a local tree in obtaining said lock, said local tree being local to the client application and having a mount point usable by the client application to lock said resource via the server data tree, said local tree comprising a data tree accessible by a plurality of threads of the multithreaded client application, and wherein said resource is further lockable via another mount point of one of said local tree and another local tree.

2. (Canceled).

3. (Currently Amended) A system of managing locking of resources of a global data repository of a distributed computing environment, said system comprising:

means for issuing a request, via a thread of a multithreaded client application of said distributed computing environment, for a lock of a resource associated with a server data tree of said distributed global data repository; and

means for obtaining said lock for said ~~thread~~ multithreaded client application, said lock being independent of a threading model of ~~an operating system~~ the requesting thread of the multithreaded client application of said distributed computing environment,

wherein said means for obtaining comprises means for employing a local tree in obtaining said lock, said local tree being local to the client application and having a mount point usable by the client application to lock said resource via the server data tree, said local tree comprising a data tree accessible by a plurality of threads of the multithreaded client application, and wherein said resource is further lockable via another mount point of one of said local tree and another local tree.

4. (Canceled).

5. (Currently Amended) At least one program storage device readable by a machine, tangibly embodying at least one program of instructions executable by the machine to perform a method of managing locking of resources of a global data repository of a distributed computing environment, said method comprising:

issuing a request, via a thread of a multithreaded client application of said distributed computing environment, for a lock of a resource associated with a server data tree of said global data repository; and

obtaining said lock for said ~~thread~~ multithreaded client application, said lock being independent of a threading model of ~~an operating system~~ the requesting thread of the multithreaded client application of said distributed computing environment,

wherein said obtaining comprises employing a local tree in obtaining said lock, said local tree being local to the client application and having a mount point usable by the client application to lock said resource via the server data tree, said local tree comprising a data tree accessible by a plurality of threads of the multithreaded client application, and wherein said resource is further lockable via another mount point of one of said local tree and another local tree.

6. (Canceled).

7. (Canceled).

8. (Previously Presented) The method of claim 7, wherein the connecting comprises connecting the local tree to the server data tree via a mount point on the local tree.

9. (Previously Presented) The method of claim 1, wherein the issuing comprises issuing a request for a lock of at least one table of the global data repository.

10. (Previously Presented) The method of claim 1, wherein the issuing comprises issuing the request from a server associated with said resource.

11. (Previously Presented) The method of claim 1, further comprising unlocking said resource by the thread of the multithreaded client application.

12. (Previously Presented) The method of claim 1, further comprising using said resource by another thread of the multithreaded client application.

13. (Canceled).

14. (Previously Presented) The system of claim 13, wherein the means for connecting comprises means for connecting the local tree to the server data tree via a mount point on the local tree.

15. (Previously Presented) The system of claim 3, wherein the means for issuing comprises means for issuing a request for a lock of at least one table of the global data repository.

16. (Previously Presented) The system of claim 3, wherein the means for issuing comprises means for issuing the request from a server associated with said resource.

17. (Previously Presented) The system of claim 3, further comprising means for unlocking said resource by the thread of the multithreaded client application.

18. (Previously Presented) The system of claim 3, further comprising means for using said resource by another thread of the multithreaded client application.

19. (Canceled).

20. (Previously Presented) The at least one program storage device of claim 19, wherein the connecting comprises connecting the local tree to the server data tree via a mount point on the local tree.

21. (Previously Presented) The at least one program storage device of claim 5, wherein the issuing comprises issuing a request for a lock of at least one table of the global data repository.

22. (Previously Presented) The at least one program storage device of claim 5, wherein the issuing comprises issuing the request from a server associated with said resource.

23. (Previously Presented) The at least one program storage device of claim 5, further comprising unlocking said resource by the thread of the multithreaded client application.

24. (Previously Presented) The at least one program storage device of claim 5, further comprising using said resource by another thread of the multithreaded client application.

* * * * *